```
/*
Display.h - A simple track GPS to SD card logger. Display module.
TinyTrackGPS v0.9

Copyright © 2019-2021 Francisco Rafael Reyes Carmona.
All rights reserved.

rafael.reyes.carmona@gmail.com

  This file is part of TinyTrackGPS.

  TinyTrackGPS is free software: you can redistribute it and/or modify
  it under the terms of the GNU General Public License as published by
  the Free Software Foundation, either version 3 of the License, or
  (at your option) any later version.

  TinyTrackGPS is distributed in the hope that it will be useful,
  but WITHOUT ANY WARRANTY; without even the implied warranty of
  MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
  GNU General Public License for more details.

  You should have received a copy of the GNU General Public License
  along with TinyTrackGPS.  If not, see <https://www.gnu.org/licenses/>.
*/

#if ARDUINO >= 100
  #include "Arduino.h"
#else
  #include "WProgram.h"
#endif

#ifndef Display_h
#define Display_h

#include "config.h"

#if defined(DISPLAY_TYPE_LCD_16X2)
    #include <LiquidCrystal.h>
#elif defined(DISPLAY_TYPE_LCD_16X2_I2C)
    #include <LiquidCrystal_I2C.h>
#elif defined(DISPLAY_TYPE_SDD1306_128X64)
    #define U8X8_HAVE_HW_I2C
    #include <U8x8lib.h>
    //#include <U8g2lib.h>
#endif

enum Display_Type {
    SDD1306_128X64,     // Para usar pantalla OLED 0.96" I2C 128x64 pixels
    LCD_16X2,           // Para usar LCD 16 x 2 carateres.
    LCD_16X2_I2C        // Para usar LCD 16 x 2 carateres. I2C.
};

class Display {
    private:
        //byte _offset;
        byte _width;       // Width pixels or numbers of columns for LCD.
        byte _height;       // Height pixels os numbers of rows for LCD.
        Display_Type _screen;
        #if defined(DISPLAY_TYPE_LCD_16X2)
            LiquidCrystal* lcd;
```

```
61          #elif defined(DISPLAY_TYPE_LCD_16X2_I2C)
62              LiquidCrystal_I2C* lcd;
63          #elif defined(DISPLAY_TYPE_SDD1306_128X64)
64              //U8G2_SSD1306_128X64_NONAME_1_HW_I2C* u8g2_SSD1306;
65              U8X8_SSD1306_128X64_NONAME_HW_I2C* u8x8_SSD1306;
66          #elif defined(DISPLAY_TYPE_HX1230_96X68)
67              U8G2_HX1230_96X68_1_3W_SW_SPI* u8g2_HX1230;
68          #endif
69
70      public:
71          Display(Display_Type t = SDD1306_128X64);
72          Display() = delete;                                 // Constructor por defecto.
73          Display(const Display&) = delete;                   // Constructor de copia.
74
75          void start();
76          void clr();
77          void print(int, int, const char[]);
78          void print(int, const char[]);
79          void print(const char[]);
80          void print(const char[], const char[]);
81          void print(const char[], const char[], const char[]);
82          void print(const char[], const char[], const char[], const char[]);
83          void wait_anin(unsigned int);
84          void draw_wait(byte);
85          void print_PChar(byte);
86          //void splash(int time_delay = 750);
87          Display_Type display_type(){return _screen;};
88  };
89
90  #endif
```